

# **A HIGHLY-SCALABLE AUTOMATED RODENT TRAINING PLATFORM**

A Thesis  
Presented to  
The Academic Faculty

by

Yahia H. Ali

In Partial Fulfillment  
of the Requirements for the Degree  
Bachelor of Science in the  
Wallace H. Coulter Department of Biomedical Engineering

Georgia Institute of Technology  
May 2019

# **A HIGHLY-SCALABLE AUTOMATED RODENT TRAINING PLATFORM**

Approved by:

Dr. Chethan Pandarinath, Advisor  
Department of Biomedical Engineering  
*Georgia Institute of Technology and Emory  
University*

Dr. Bilal Haider  
Department of Biomedical Engineering  
*Georgia Institute of Technology and Emory  
University*

Date Approved: May 2, 2019

<b>Abstract</b>	<b>4</b>
<b>Keywords</b>	<b>4</b>
<b>Introduction</b>	<b>4</b>
<b>Materials and Methods</b>	<b>6</b>
Design Requirements	6
System Overview	7
Server Processes	7
Rodent Training	7
System Validation	9
Subjects	10
<b>Results</b>	<b>11</b>
<b>Discussion</b>	<b>13</b>
<b>Conclusions</b>	<b>14</b>
<b>Acknowledgments</b>	<b>14</b>

## **Abstract**

*Background:* Rodent training is a necessary but time-consuming process that often requires the development of computer-based training systems to automate the process of administering a food or water reward in return for the rodent performing the desired behavioral task. To increase the throughput of these systems, they need to be scaled up to simultaneously train more rodents, but current scalable automated systems are incompatible with the graphical programming software used to develop behavioral tasks.

*New Method:* Here, we present a novel scalable rodent training platform that allows researchers to scale behavioral tasks developed in graphical programming environments such as Simulink and LabVIEW. This system communicates with training cages over a network, so it is compatible with all internet-enabled devices regardless of the software or operating system they are running.

*Results:* The system is validated by training a cohort of four previously-untrained rats over a period of three weeks in a fully automated fashion. In ten sessions or less, all of the rats learned to extend to touch a knob in return for a food reward.

*Comparison with Existing Method(s):* The use of this training system to control two training cages running Simulink software and automatically adjust the training parameters according to each rat's performance represents an advance over previous single-cage training systems.

*Conclusions:* In this study, the ability to train rats on a novel forelimb perturbation task demonstrates the complex behavioral tasks that can now be studied in a scalable, automated fashion while maintaining compatibility with the graphical programming tools currently in use.

## **Keywords**

high-throughput, rodent, rat, automated, training, neuroscience, kinematics, behavioral, real-time, controls, conditioning

## **Introduction**

In neuroscience, rodents are often used as a model for investigating the way in which the brain controls behavior. They are an appealing model to use because of their low cost, small footprint, and extensive reference literature. However, training these rodents to perform the behaviors necessary for each experiment is a time-consuming and challenging process. This makes rodent studies difficult to scale, and it inhibits scientists from studying those questions that would benefit from a large sample size.

To overcome the limitations of rodent training, a number of strategies have been developed that reduce the amount of time necessary for a researcher to spend manually training each animal, while also allowing researchers to investigate more complicated rodent behaviors. A common solution is to develop a training cage for the rodent that distributes a reward when the rodent performs a desirable behavior. Each cage is typically specially designed for the task being studied, and, with this approach, researchers have created training cages that span a variety of

different rodent tasks (Table 1). Often this type of system still requires the researcher to initiate and end the rodent's training sessions, so others have developed systems that use a centralized server to interface with many automated training cages at once, allowing these cages to execute training sessions on a predetermined schedule (Poddar et al., 2013). The main advantage of this approach is that it offers the ability to train multiple rodents in parallel on the same types of tasks that can be conducted in standard automated training cages. These tasks include lever presses and forelimb rotation, which are commonly used in neuroscientific literature.

Many existing automated training systems are well-designed for the tasks that they must execute, but they are challenging to scale due to their lack of a centralized manner of controlling multiple training cages. At the same time, it is difficult to incorporate these single-cage systems into broader multi-cage systems like that presented by Poddar et al in 2013 because of restrictions of the type of software that these multi-cage systems can interface with. In the case of the system used in Poddar et al 2013, behavioral tasks had to be programmed in C# or another .NET programming language, which are not widely used for other behavioral tasks. This issue demonstrates a need for a scalable training platform that is broadly compatible with the software used with most single-cage training systems, which is often developed in graphical programming environments such as LabVIEW and Simulink.

Scaling single-cage training systems can now be achieved using RatCoach, a modular automated training system that controls training cages via standard networking protocols that are compatible with all internet-enabled computers and microcontrollers. The RatCoach system can be used with no knowledge of the software or operating system of the training cage computers that it controls because it executes all interactions with training cages at the network level. This means that any internet-enabled training cage can be used with RatCoach with only minor modifications to the original training cage code.

Reference	Task	Hardware	Software Language	Compatible with UDP?
(Vigaru et al., 2013)	reaching with a perturbable 3-DOF robotic manipulandum	desktop computer and NI PCI-6221 card	LabView	Yes
(Poddar et al., 2013)	timed lever pressing	desktop computer and NI PCIe 6323	C#, .NET	Yes
(Wong et al., 2015)	pellet-grasping	desktop computer and Arduino	MATLAB	Yes
(Meyers et al., 2016)	forelimb rotation	desktop computer	MATLAB	Yes
(Ellens et al., 2016)	pellet-grasping	desktop computer and NI PCI cards	LabView	Yes
(Qiao et al., 2018)	selective attention task	desktop computer or Raspberry Pi	MATLAB or Python	Yes
(Bollu et al., 2019)	hold-still center-out reach task	NI sbRIO-9636	LabView	Yes

**Table 1. Automated rodent training tasks.** A number of automated rodent training tasks have been developed. These tasks differ not only in the behavior that they test, but also in the hardware and software on which they are built. National Instruments (NI) data acquisition systems are widely used, and most systems include a desktop computer. All of the systems reviewed here are internet-enabled and thus compatible with network communication via the User Datagram Protocol (UDP) (Postel, 1980).

## Materials and Methods

### *Design Requirements*

Based on a review of current automated training cage systems (Table 1), it was determined that RatCoach should be compatible with behavioral tasks programmed in Mathworks' MATLAB and Simulink and National Instruments' LabVIEW. RatCoach should also be written in a language commonly used by neuroscientists in order to encourage collaboration and customization of the code to each individual's use case.

In addition to the technical requirements of the system, RatCoach would also need to automate four actions commonly performed in rodent behavioral training: 1) running training sessions on a schedule, 2) logging training results, 3) evaluating each rat's performance, and 4) adjusting the training task. This automation was accomplished by a set of processes running on the RatCoach server.

### *System Overview*

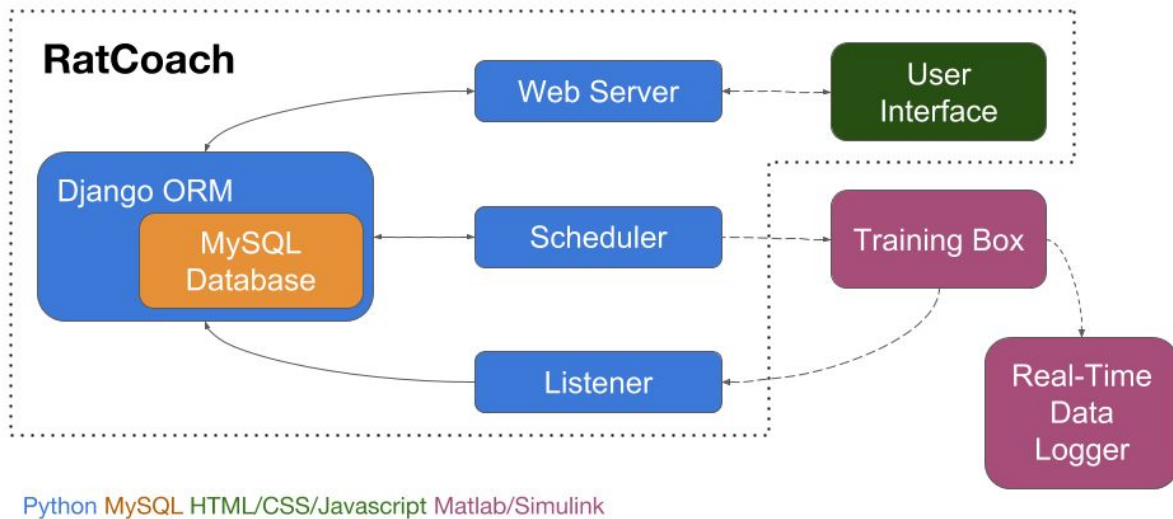
With the RatCoach system, training cages continuously send to and receive data from the RatCoach server. This server controls the scheduling of training sessions and the progression of each rodent's training. It can be accessed and controlled via a self-hosted website in which the user can view the performance of each rodent and make adjustments to their training when necessary. The user can also configure RatCoach to automatically change training cage parameters until the desired level of rodent performance is reached.

### *Server Processes*

RatCoach has one database and three server-side processes that together, allow the user to control the training being performed by a given set of training cages. The first of these processes is the Django web server, which has been used without modification to serve the RatCoach web application to the user. The second process, named "Scheduler", starts and stops training sessions according to a user-defined schedule. Scheduler is also responsible for evaluating each rat's performance and adjusting the training task accordingly at the start of each training session. The third process, called "Listener", receives network packets from training cages and logs them into the database. These packets can either contain trial data or status messages about each training cage. This trial data packet does not include time series data such as kinematic trajectories. These are instead sent to a real-time data logger separate from the RatCoach system (Figure 1).

### *Rodent Training*

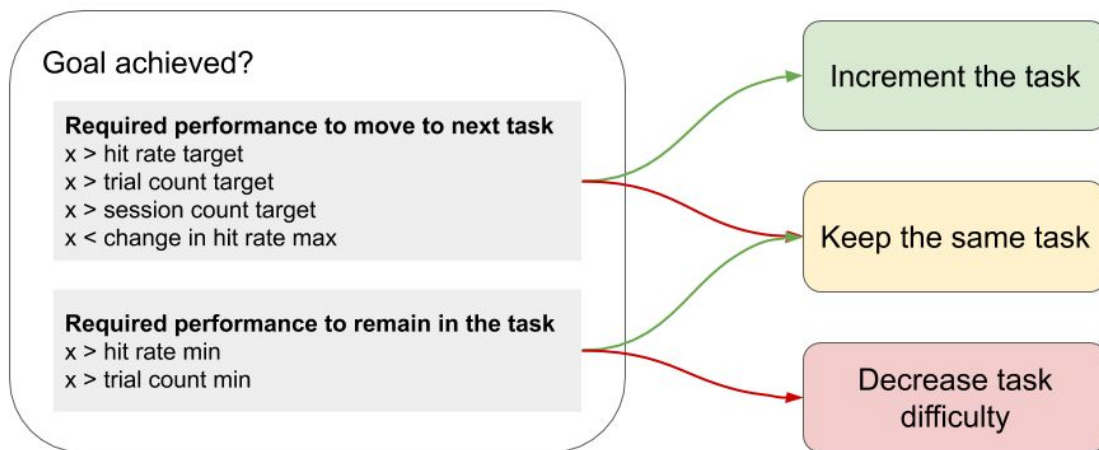
During the course of a rodent training session, RatCoach records packets of data summarizing the rodent's performance on each trial. To allow for customization, RatCoach places no restrictions on the contents of this data packet except that it must contain a field named "hit" that includes values of "1" or "0" depending on whether the rodent succeeded or failed in the trial. Other fields such as "turn angle" and "hold time" can be included for later analysis, but they will not be automatically analyzed by RatCoach for task adjustment purposes.



**Figure 1. Layout of the RatCoach system.** The information RatCoach uses to track rodent training is stored in a MySQL data that is accessed via the object-relational mapping (ORM) included with the Django Python library. The Django library also includes a web server that displays a user interface programmed in HTML, CSS, and Javascript. Scheduler is a process that starts and stops training sessions and adjusts training parameters according to the data and settings stored in the database. Listener receives packets of data from the training boxes and stores those in the database for Scheduler to analyze at the start of each session. The training box sends trial-level data to RatCoach and millisecond-level data (e.g. kinematic data, camera images) to a real-time data logger. The dashed lines indicate communication that occurs over a network, and solid lines indicate communication that occurs via a hard disk.

From RatCoach's trial data, the system can calculate three metrics used in evaluating a rodent's performance in each training session: 1) hit rate, 2) trial count, and 3) change in hit rate. A fourth "session count" metric is also used to represent the number of sessions the rodent has completed with the current behavioral task. When specifying a task for a rodent to perform, the user can set four different thresholds based on these metrics that define the minimum performance needed to increment the task difficulty. If specified, the rat will need to reach all four of these thresholds before the task difficulty can increase. The user can also specify three different thresholds for keeping the rodent at the same level of difficulty. If any of these thresholds are not reached, the system will decrease the difficulty of the current task but will not move the rat to an earlier task. If the current task is already at its minimum difficulty level, then no change is made (Figure 2).





**Figure 2. Automatically adjusting rodent training.** Let “x” represent a summary of the subject’s performance in a previous session. RatCoach compares this performance with the user-defined thresholds for moving to the next task and for remaining in the same task. If all of the thresholds for moving to the next task are reached, then the subject progresses to the next task or to a more difficult version of the current task. If any of the thresholds required to remain in the current task are not reached, then RatCoach will decrease the difficulty of the current task if possible. Otherwise, RatCoach will keep the subject in the same task for its upcoming training session.

To simplify the process of specifying tasks for many different rodents, the user can group tasks into "paradigms". Each paradigm contains a checklist of tasks and allows the user to specify the order in which tasks must be completed. Assigning a paradigm to a rodent prompts the system to generate a list of tasks for that particular rodent that functions as a copy of those tasks plus the addition of some metrics that track the subject’s progress.

### *System Validation*

To validate RatCoach's ability to automatically train rodents, it was used with a set of two automated training cages to train four previously-untrained rats on a forelimb-reaching behavior. The user performing the training only had access to the system's graphical user interface and not to the database itself. Using this interface, the user created a forelimb-reaching paradigm containing four tasks and then assigned that paradigm to four different rats.

For the first task, RatCoach performed sound-food association for one session without recording any data on the performance of the rat. This task was simply meant to condition the rat to associate a sound tone with the administration of a food reward. The second task involved knob-food association, in which the training cage distributes a reward each time the rat touches a knob at the front of its cage. RatCoach was instructed to mark this task as complete when the rat performed 60 or more trials in a single session. For the final task, RatCoach would need to gradually increase the distance of the knob from the front of the cage, thus requiring the rat to

extend their forelimb further than before. This task tested RatCoach's ability to adjust training parameters based on performance because it had to retract the knob every time the rat completed 30 or more trials per session (Table 2).

### *Subjects*

The rats used in this validation study were socially-housed female Long Evans rats weighing 250-300 grams. They were trained twice per weekday for a total of ten 30-minute training sessions per week. They were removed from their home cage and placed into the automated training cage at the start of each session and returned to their home cage after the session was completed. Over the weekend, the rats were fed *ad libidum*, but during the weekdays they were restricted to only the food they received during training sessions. All housing and procedures were approved by the Institutional Animal Care and Use Committee at Emory University (Animal Welfare Assurance Number A3180-01).

Task	Adaptive Parameter	Description
Sound-food association	None	Play a tone while distributing food pellets
Knob-food association	None	Distribute a reward pellet when the rat touches the knob
Knob retraction	Knob distance	Gradually increase the knob's distance from the training cage wall to require the rat to reach for it
Knob holding	Minimum knob hold time	Gradually increase the minimum amount of time that the rat must hold the knob before receiving a reward

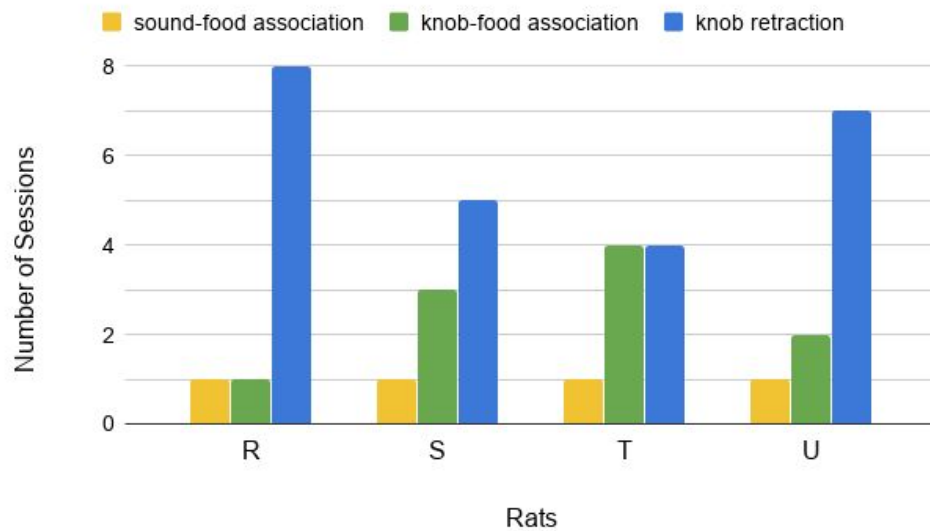
**Table 2. Task Descriptions.** The RatCoach system was validated by training four rats on a sequence of four behavioral tasks. The sound-food association and knob-food association tasks contain no adaptive parameter, so they will simply be marked as “complete” once the user-defined training goals are achieved. In the knob retraction task, the knob distance parameter will change after every session in which the rat achieves the training goal until the final knob distance value is reached. The same will happen for the knob holding task except that the parameter that will vary is the minimum knob hold time needed to receive a reward.

## Results

Following a three-week training period, the cohort of four rats completed the sound-food association, knob-food association, and knob retraction tasks in a fully automated fashion guided by RatCoach. One rat (Rat S) also completed knob-holding training, while the others were at varying stages of progressing to longer minimum knob hold times.

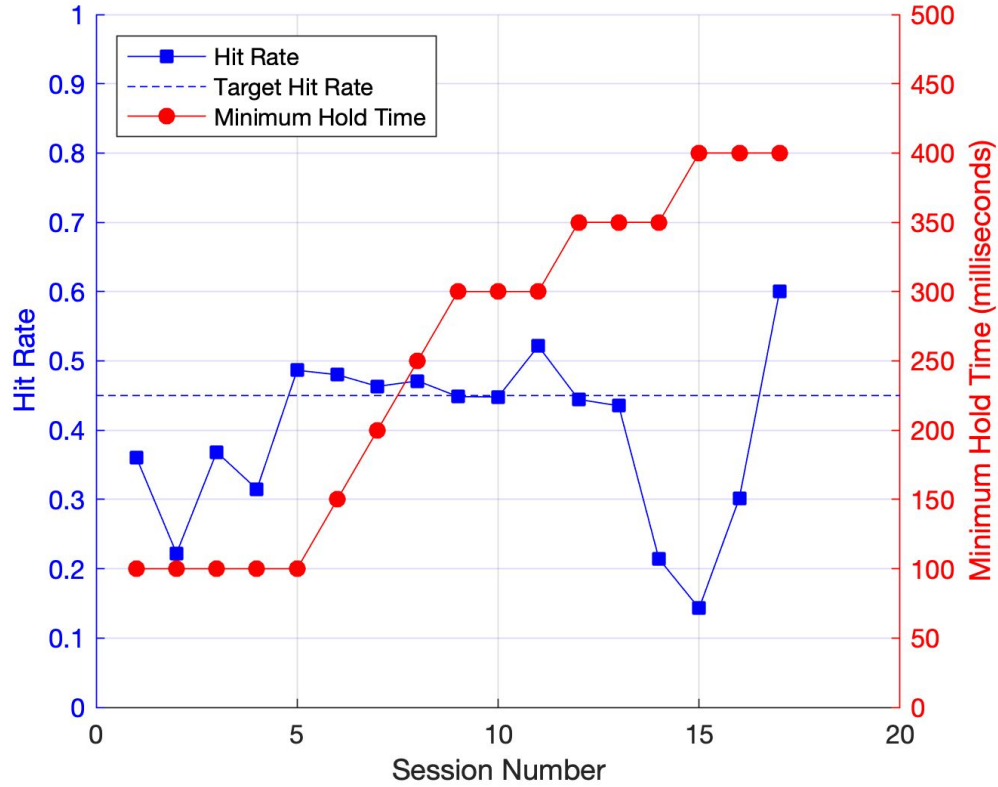
In the first task, the user set a limit of one session because the task was simply designed to have the rodent assimilate to the training cage. No performance data was recorded from this task. Following that sound-food association session, RatCoach began knob-food association. In this task, each rat learned that tapping a touch-sensitive knob would result in a reward. The desired level of behavior on this task was observed with each rat in 1-5 sessions. In knob retraction training, the rats continued to touch the knob in return for a reward, but, every session, the knob would be positioned further and further back from the cage. This parameter on the position of the knob was set automatically by RatCoach based on the range that had been specified by the user and in response to the rat's performance. In this task, all four rats were able to complete knob retraction training in 4-8 sessions. Note that this task, as it was originally

configured, would have taken a minimum of eight sessions to complete, but, upon observing Rat S could easily adapt to the small changes in knob distance, the user changed the task in the RatCoach system to use four different retraction values rather than eight (Figure 3).



**Figure 3. Rat training progress.** Using RatCoach, rats R, S, T, and U completed sound-food association, knob-food association, and knob retraction tasks. The sound-food association task was configured to last for only one session, but the progression from the knob-food association and knob retraction tasks were based on each rat's performance. All four rats were able to complete these three tasks in 9-10 sessions.

After three weeks of training, rat's R, T, and U had reached minimum hold times of 150, 200, and 100 milliseconds, respectively. Rat S had completed the knob-holding task, learning to hold the knob for 400 milliseconds before receiving a reward. Rat S progressed rapidly from a hold time of 100 milliseconds to a hold time of 300 milliseconds in four sessions and then more slowly achieved a hold time of 400 milliseconds in eight sessions (Figure 4).



**Figure 4. Knob Holding Training for Rat S.** Following the first three training tasks (Fig. 3), Rat S completed knob-holding training, which is aimed at training the rat to hold a touch-sensitive knob for 400 milliseconds or more. Beginning with a user-defined minimum hold time of 100 milliseconds, RatCoach progressively increased the minimum hold time when the rat achieved a hit rate of 0.45 or more. Note that the progression from 350 to 400 milliseconds for session 15 was triggered by a user mistakenly starting and stopping a session prematurely, resulting in an unusually high hit rate that RatCoach used to increase the minimum hold time.

## Discussion

Rodent training is a labor-intensive process that often requires frequent human supervision to ensure that the subject is performing the experiment's desired behavior. With single-cage automated training systems, the time spent closely observing each rodent may be reduced, but the researcher must spend additional time configuring each automated training cage. In this validation study, RatCoach, a system that automated the management of multiple training cages, successfully training four rats on a forelimb-reaching task based on tapping a touch-sensitive knob in return for a reward.

It should be noted that these training outcomes were not achieved without human intervention, but that intervention was limited to what could be performed using RatCoach's user interface. For instance, the researcher's adjustment of the knob retraction training to require less intermediate training steps was based on their observation that Rat S was adapting to the small

changes in knob distance very easily throughout the knob retraction stage. This level of human supervision is expected, but it may be something that future automated systems could further minimize by better configuring the training task using prior knowledge of training outcomes.

The time-savings of the RatCoach system were not evaluated in this study, but the training results demonstrated the advantage of having a system that automates task adjustment for each individual subject. The rats were shown to learn the behavioral tasks at different rates, so being able to automatically act on each individual rat's performance data reduced the amount of time the user needed to spend monitoring each rat. For example, when Rat R completed the knob-food association task faster than the other rats, RatCoach could automatically move Rat R to the next task without any input from the user. This may not save much time for a system of four rats, but when training larger populations it is expected that this would greatly speed up training.

## **Conclusions**

We have developed and validated an automated training system that adjusts training parameters based on each subject's prior performance. While this system was validated on a single task, RatCoach has been designed in a way that allows it to interface with any internet-enabled training cage. In future work, validating this system with different behavioral tasks and larger numbers of subjects could allow for the optimization of behavioral task adjustment using machine learning models that leverage the knowledge of prior training performance. Automated training methods such as RatCoach will be key to developing such systems and continuing to make rodent training more accessible to researchers.

## **Acknowledgments**

This work was supported by funding from the Wallace H. Coulter Department of Biomedical Engineering at Georgia Tech and Emory University and the Georgia Institute of Technology's Undergraduate Research Opportunities Program. I would also like to thank Dilara Soyly, Jordan Kruguer, Valentine Wilson, and Yaxuan Wang for helping to develop RatCoach and Feng Zhu for helping to conduct its validation study.

## References

- Bollu, T., Whitehead, S.C., Prasad, N., Walker, J., Shyamkumar, N., Subramaniam, R., Kardon, B., Cohen, I., Goldberg, J.H., 2019. Automated home cage training of mice in a hold-still center-out reach task. *Journal of Neurophysiology*. <https://doi.org/10.1152/jn.00667.2018>
- Ellens, D.J., Gaidica, M., Toader, A., Peng, S., Shue, S., John, T., Bova, A., Leventhal, D.K., 2016. An automated rat single pellet reaching system with high-speed video capture. *J. Neurosci. Methods* 271, 119–127. <https://doi.org/10.1016/j.jneumeth.2016.07.009>
- Meyers, E., Sindhurakar, A., Choi, R., Solorzano, R., Martinez, T., Sloan, A., Carmel, J., Kilgard, M.P., Rennaker, R.L., 2nd, Hays, S., 2016. The supination assessment task: An automated method for quantifying forelimb rotational function in rats. *J. Neurosci. Methods* 266, 11–20. <https://doi.org/10.1016/j.jneumeth.2016.03.007>
- Poddar, R., Kawai, R., Ölveczky, B.P., 2013. A fully automated high-throughput training system for rodents. *PLoS One* 8, e83171. <https://doi.org/10.1371/journal.pone.0083171>
- Postel, J., 1980. User Datagram Protocol (No. RFC 768), Internet Requests for Comment. RFC Editor. <https://doi.org/10.17487/RFC0768>
- Qiao, M., Zhang, T., Segalin, C., Sam, S., Perona, P., Meister, M., 2018. Mouse Academy: high-throughput automated training and trial-by-trial behavioral analysis during learning. *Animal Behavior and Cognition*. <https://doi.org/10.1101/467878>
- Vigaru, B.C., Lamercy, O., Schubring-Giese, M., Hosp, J.A., Schneider, M., Osei-Atiemo, C., Luft, A., Gassert, R., 2013. A robotic platform to assess, guide and perturb rat forelimb movements. *IEEE Trans. Neural Syst. Rehabil. Eng.* 21, 796–805. <https://doi.org/10.1109/TNSRE.2013.2240014>
- Wong, C.C., Ramanathan, D.S., Gulati, T., Won, S.J., Ganguly, K., 2015. An automated behavioral box to assess forelimb function in rats. *J. Neurosci. Methods* 246, 30–37. <https://doi.org/10.1016/j.jneumeth.2015.03.008>